

**KOREAN INDUSTRIAL
PROPERTY OFFICE**

This is to certify that the following application annexed hereto
is a true copy from the records of the Korean Industrial
Property Office.

Application Number: Patent Application No. 2001-40704

Date of Application: 7 July 2001

Applicant(s): Samsung Electronics Co., Ltd.

16 August 2001

COMMISSIONER

1020010040704

2001/8/17

[Document Name] Patent Application

[Application Type] Patent

[Receiver] Commissioner

[Reference No.] 0007

[Filing Date] 2001.07.07

[IPC] G06F

[Title] Methods and apparatus for compression and restoration of animation path using linear approximation and data format therefor

[Applicant]

[Name] Samsung Electronics Co., Ltd.

[Applicant code] 1-1998-104271-3

[Attorney]

[Name] Young-pil Lee

[Attorney's code] 9-1998-000334-6

[General Power of Attorney Registration No.] 1999-009556-9

[Attorney]

[Name] Hae-young Lee

[Attorney's code] 9-1999-000227-4

[General Power of Attorney Registration No.] 2000-002816-9

[Inventor]

[Name] JANG, Eeu Seon

[I.D. No.] 681220-1481119

[Zip Code] 442-470

[Address] 542-904 Shinnamushil Ssangyong Apt., Youngtong-dong
Paldal-gu, Suwon-city, Kyungki-do

[Nationality] Republic of Korea

[Inventor]

[Name] KIM, Do Kyoon

1020010040704

2001/8/17

[I.D. No.] 690605-1041815
[Zip Code] 463-030
[Address] 3rd Floor, 175-1 Bundang-dong, Bundang-gu
Seongnam-city, Kyungki-do
[Nationality] Republic of Korea

[Inventor]

[Name] WOO, Sang Oak
[I.D. No.] 741211-1046711
[Zip Code] 121-041
[Address] 111-302 Hyundai Apt., Dohwa 1-dong
Mapo-gu, Seoul
[Nationality] Republic of Korea

[Priority Claimed]

[Application Country] Republic of Korea
[Type of Application] Patent
[Application No.] 10-2000-0070090
[Filing Date] 2000.11.23
[Priority Document] Attached

[Application Order] I/We file as above according to Art. 42 of the Patent Law.
Attorney Young-pil Lee
Attorney Hae-young Lee

[Fee]

[Basic page]	20 Sheet(s)	29,000 won
[Additional page]	25 Sheet(s)	25,000 won
[Priority claiming fee]	1 Case(s)	26,000 won
[Examination fee]	0 Claim(s)	0 won
[Total]	80,000 won	

[Enclosures]

1. Abstract and Specification (and Drawings)_1 copy

대한민국 특허청

KOREAN INTELLECTUAL PROPERTY OFFICE

JCS21 U.S. PTO
09/988708
11/20/01

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto
is a true copy from the records of the Korean Intellectual
Property Office.

출원 번호 : 특허출원 2001년 제 40704 호
Application Number PATENT-2001-0040704

출원 년 월 일 : 2001년 07월 07일
Date of Application JUL 07, 2001

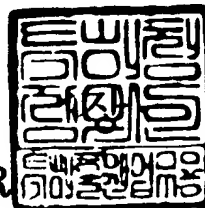
출원인 : 삼성전자 주식회사
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.

CERTIFIED COPY OF
PRIORITY DOCUMENT

2001 년 08 월 16 일

특 허 청

COMMISSIONER



【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0007
【제출일자】	2001.07.07
【국제특허분류】	G06F
【발명의 명칭】	선형 근사화를 이용한 애니메이션 궤적의 압축 및 복원 방법들, 이 방법들을 수행하는 장치들 및 이를 위한 데이터 포맷
【발명의 영문명칭】	Methods and apparatus for compression and restoration of animation path using linear approximation and data format therefor
【출원인】	
【명칭】	삼성전자 주식회사
【출원인코드】	1-1998-104271-3
【대리인】	
【성명】	이영필
【대리인코드】	9-1998-000334-6
【포괄위임등록번호】	1999-009556-9
【대리인】	
【성명】	이해영
【대리인코드】	9-1999-000227-4
【포괄위임등록번호】	2000-002816-9
【발명자】	
【성명의 국문표기】	장의선
【성명의 영문표기】	JANG,Eeu Seon
【주민등록번호】	681220-1481119
【우편번호】	442-470
【주소】	경기도 수원시 팔달구 영통동 신나무실 쌍용아파트 542동 904호
【국적】	KR
【발명자】	
【성명의 국문표기】	김도균
【성명의 영문표기】	KIM,Do Kyoon

【주민등록번호】	690605-1041815
【우편번호】	463-030
【주소】	경기도 성남시 분당구 분당동 175-1 3층
【국적】	KR
【발명자】	
【성명의 국문표기】	우상옥
【성명의 영문표기】	W00,Sang Oak
【주민등록번호】	741211-1046711
【우편번호】	121-041
【주소】	서울특별시 마포구 도화1동 현대아파트 111동 302호
【국적】	KR
【우선권주장】	
【출원국명】	KR
【출원종류】	특허
【출원번호】	10-2000-0070090
【출원일자】	2000.11.23
【증명서류】	첨부
【취지】	특허법 제42조의 규정에 의하여 위와 같이 출원합니다. 대리인 이영필 (인) 대리인 이해영 (인)
【수수료】	
【기본출원료】	20 면 29,000 원
【가산출원료】	25 면 25,000 원
【우선권주장료】	1 건 26,000 원
【심사청구료】	0 항 0 원
【합계】	80,000 원
【첨부서류】	1. 요약서·명세서(도면)_1통

【요약서】**【요약】**

선형 근사화를 이용한 애니메이션 궤적의 압축 및 복원 방법들, 이 방법들을 수행하는 장치들 및 이를 위한 데이터 포맷이 개시된다. 이 압축 장치는, 원래의 애니메이션 궤적으로부터 분절점들을 추출하는 보간 분석부와, 보간 분석부로부터 출력되는 분절점들을 이용하여, 보간 분석부로부터 입력한 키를 부호화하는 키 부호화부와, 보간 분석부로부터 출력되는 분절점들을 이용하여, 보간 분석부로부터 입력한 키값, 키값의 최소값 및 최대값들을 부호화하는 키값 부호화부 및 키 및 키값 부호화부들에서 부호화된 결과들을 엔트로피 부호화하여, 부호화된 비트 스트림으로서 출력하는 엔트로피 엔코더를 구비하는 것을 특징으로 한다.

【대표도】

도 5a

【명세서】**【발명의 명칭】**

선형 근사화를 이용한 애니메이션 궤적의 압축 및 복원 방법들, 이 방법들을 수행하는 장치들 및 이를 위한 데이터 포맷{Methods and apparatus for compression and restoration of animation path using linear approximation and data format therefor}

【도면의 간단한 설명】

도 1은 일반적인 삼차원 애니메이션에서 애니메이션 궤적을 설명하기 위한 도면이다.

도 2는 가상 현실 모델링 언어 또는 엠펙-4에서 사용되는 애니메이션 궤적의 표현 예를 나타낸다.

도 3은 삼차원 애니메이션의 데이터 형식을 설명하기 위한 도면이다.

도 4a 및 4b들은 애니메이션 궤적에 대한 종래의 압축 장치 및 복원 장치의 블록도들이다.

도 5a 및 5b들은 본 발명에 의한 애니메이션 궤적의 압축 장치 및 복원 장치의 블록도들이다.

도 6은 도 5a에 도시된 압축 장치의 본 발명에 의한 바람직한 일실시예의 블록도이다.

도 7은 도 5b에 도시된 복원 장치의 본 발명에 의한 바람직한 일실시예의 블록도이다.

도 8a-8h들은 선형 근사화를 이용한 분절점 추출의 본 발명에 의한 바람직한 일 실시예를 설명하기 위한 도면들이다.

도 9는 실제 애니메이션 궤적과 근사화된 애니메이션 궤적의 차이를 구하는 방법을 설명하기 위한 도면이다.

도 10은 양자화 과정을 나타내는 도면으로서, 일반적으로 사용되는 DPCM 부호화 방식을 나타낸다.

도 11a-11e들은 본 발명에 의한 부호화된 비트 스트림의 포맷을 나타내는 도면이다.

도 12는 문법표현으로 바꾸어 표현한 본 발명에 의한 복호화 처리 과정을 설명하기 위한 도면이다.

도 13a-13d들은 본 발명에 의한 압축 방법과 종래의 압축 방법을 비교하기 위한 실험 애니메이션 시퀀스에 대한 그래프들이다.

도 14a-14f는 선형 근사화를 이용한 분절점 추출의 본 발명에 의한 바람직한 또 다른 일 실시예를 설명하기 위한 도면들이다.

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

<15> 본 발명은 3차원 그래픽스 모델의 애니메이션(animation)에 관한 것으로서, 특히 애니메이션에 사용되는 애니메이션 궤적을 선형 근사화를 이용하여 압축 및

복원하는 장치들, 이 장치들에서 수행되는 압축 및 복원 방법들 및 이를 위한 데이터의 포맷에 관한 것이다.

<16> 그래픽스에 기반한 삼차원 애니메이션의 경우, 3차원 모델로 표현된 객체의 애니메이션은 공간상의 움직임, 회전, 모델 자체의 모핑(morphing), 색의 변이 등을 표현하기 위하여 인터폴레이터(interpolator)를 사용한다.

<17> 도 1은 일반적인 삼차원 애니메이션에서 애니메이션 궤적을 설명하기 위한 도면으로서, 종축은 키 값(KEY_VALUE)을 나타내고, 횡축은 키(KEY)를 각각 나타낸다.

<18> 도 1에 도시된 바와 같이, 애니메이션 궤적(20)은 3차원 모델(10)의 애니메이션될 자취를 표현하는 것이다. 각 애니메이션 정보의 궤적(20)은 도 1에 도시된 바와 같이 시간에 따라 변화하는 2차원 곡선의 형태를 갖는다. 애니메이션 궤적(20)을 표현하는 방식은 여러가지가 있으며, 이러한 여러가지 방식들은 1989년에 A. K. Jain에 의해 'Fundamentals of Digital Image Processing'라는 제목으로 Prentice Hall 출판사에 의해 출간된 책의 Chapter 9에 개시되어 있다.

<19> 인터폴레이터 표현에서는 도 1에 도시된 바와 같이 곡선 형태의 애니메이션 궤적(20)을 분절 가능한 여러개의 직선형태로 나누어 표현할 수 있다. 이러한 표현에서 반드시 필요한 정보는 각 직선의 분절 점(break point) 또는 꼭지점이다. 여기서, 분절점 또는 꼭지점은 도 1에 도시된 애니메이션 궤적(20)에서 점으로 표시되어 있다. 선형 삽입(Linear Interpolation)을 이용하면 각 분절 점으로부터 원래의 곡선을 복원할 수 있다.

<20> 도 2는 가상 현실 모델링 언어(VRML:Virtual Reality Modeling Language) 또는 엠펙(MPEG:Moving Picture Expert Group)-4에서 사용되는 애니메이션 궤적의 표현 예(ScalarInterpolator)를 나타낸다.

<21> 인터플레이터의 표현예는 도 2에 도시된 바와 같다. 처리할 정보는 키(Key) 및 키값(Key Value)이 되고, 주어진 정보로부터 선형 삽입 작업을 하도록 되어 있다. 인터플레이터의 종류는 크게 6가지 즉, 스칼라 인터플레이터, 위치 인터플레이터, 좌표 인터플레이터, 회전 인터플레이터, 법선 인터플레이터 및 색 인터플레이터들이 있다. 여기서, 스칼라 인터플레이터는 도 2에 도시된 바와 같이 표현될 수 있다. 각 인터플레이터의 종류, 특징 및 기능은 다음 표 1에 표기되어 있지만, 기본적으로 주어진 키와 그에 따른 키 값의 집합이라는 점에서 일치한다

<22> 【표 1】

종 류	특 징	기 능
스칼라 인터플레이터	스칼라 변화량의 선형 삽입	넓이, 반경, 강도 등의 표현
위치 인터플레이터	3차원 좌표상의 선형 삽입	3차원 공간상의 평행 이동
회전 인터플레이터	3차원 좌표축과 회전량의 선형	3차원 공간상의 회전
좌표 인터플레이터	3차원 모델의 좌표 변화량과	3차원 모핑(morphing)
법선 인터플레이터	3차원 법선 벡터의 선형 삽입	3차원 법선 벡터의 변화 표현
색 인터플레이터	색상 정보의 선형 삽입	색상 변화량의 표현

<23> 도 3은 삼차원 애니메이션의 데이터 형식을 설명하기 위한 도면으로서, 엔코더(encoder)(30), 디코더(decoder)(40) 및 3차원 애니메이션 파일 포맷(50)으로 구성된다. 여기서, 엔코더(30)로부터 디코더(40)로 출력되는 3차원 애니메이션 파일 포맷(50)은 모델 데이터(MODEL DATA), 애니메이션 데이터(ANIMATION

DATA), 속성들(ATTRIBUTES), 비디오/텍스처(VIDEO/TEXTURE) 및 사운드(SOUND)로 구성된다.

<24> 도 3을 참조하면, 인터플레이터는 삼차원 애니메이션 궤적을 효과적으로 표현하는 애니메이션 데이터에 해당한다. VRML/MPEG-4 등으로 표현되는 삼차원 애니메이션 데이터는 도 3에 도시된 바와 같은 정보들로 구성된다. 종래의 오디오, 비디오, 3차원 모델의 경우 표준화된 압축 기술이 제공되는 반면에, 애니메이션 궤적을 결정하는 인터플레이터의 경우에는 표현 위주의 범용 압축 기술만이 제공되고 있다. 오디오/비디오를 뺀 애니메이션의 경우, 애니메이션 궤적 데이터량은 3차원 모델과 함께 대부분을 차지하게 된다. 따라서, 3차원 모델의 압축 기술과 함께 애니메이션 궤적 압축 기술이 필수적이다. MPEG-4 BIFS(Binary Format for Scene)에서 기본적인 양자화/압축을 제공하지만, 이는 인터플레이터를 위한 압축이 아닌 범용 압축 기술로서 취약한 압축 성능을 갖는 문제점이 있다. 이에 대해서는, 2000년 New York(NY)에서 열린 멀티미디어 및 엑스포에 대한 국제 회의(international Conference on multimedia and Expo)에서 Euee S. Jang에 의해 '3D Animation Coding : its History and Framework'라는 제목으로 발표된 회보(proceedings)에 개시되어 있다.

<25> 도 4a 및 4b들은 애니메이션 궤적에 대한 종래의 압축 장치 및 복원 장치의 블럭도들로서, 도 4a에 도시된 종래의 압축 장치는 스칼라 양자화(scalar quantization)부(60)로 구성되고, 도 4b에 도시된 종래의 복원 장치는 스칼라 역양자화(scalar dequantization)부(70)로 구성된다.

<26> 도 4a에 도시된 스칼라 양자화부(60)는 원래의 애니메이션 경로(path)를 (key, key_value)(여기서, key는 키를 나타내고, key_value는 키 값을 각각 나타낸다.)의 형태로 입력단자 IN1을 통해 입력하여 스칼라 양자화하고, 스칼라 양자화된 결과인 부호화된 비트 스트림을 출력단자 OUT1을 통해 출력한다. 도 4b에 도시된 스칼라 역 양자화부(70)는 부호화된 비트 스트림을 입력단자 IN2를 통해 입력하고, 복원된 애니메이션 경로를 (key, key_value)의 형태로 출력단자 OUT2를 통해 출력한다.

<27> 결국, 종래의 MPEG-4 BIFS에서 인터폴레이터의 압축은 도 4a에 도시된 바와 같이 스칼라 양자화를 거친다. 도 4a에 도시된 종래의 압축 과정은 인터폴레이터 뿐만 아니라 BIFS상에서 압축을 요구하는 모든 요소들에 범용으로 제공되는 방식이다. 도 4b에 도시된 종래의 복원 장치에 입력된 부호화된 비트 스트림은 압축의 역순에 의해 스칼라 역 양자화부(70)를 거쳐 애니메이션 경로로 복원된다. 도 4a 및 4b에 도시된 바와 같이, 인터폴레이터에서 제공되는 키 및 키 값등이 각 종류별 특성이 고려되지 않고, 동일한 방식으로 압축되므로, 충분한 압축 효과를 가져오지 못하는 문제점이 있다.

【발명이 이루고자 하는 기술적 과제】

<28> 본 발명이 이루고자 하는 제1 기술적 과제는, 인터폴레이터 형식의 애니메이션 데이터를 효과적으로 압축하여 빠른 데이터 전송 및 저장을 가능하게 하는 선형 근사화를 이용한 애니메이션 궤적의 압축 장치를 제공하는 데 있다.

<29> 본 발명이 이루고자 하는 제2 기술적 과제는, 상기 제1 기술적 과제에 의해 압축된 데이터를 효과적으로 복원하는 선형 근사화를 이용한 애니메이션 궤적의 복원 장치를 제공하는 데 있다.

<30> 본 발명이 이루고자 하는 제3 기술적 과제는, 상기 제1 기술적 과제에 의한 선형 근사화를 이용한 애니메이션 궤적의 압축 장치에서 수행되는 애니메이션 궤적의 압축 방법을 제공하는 데 있다.

<31> 본 발명이 이루고자 하는 제4 기술적 과제는, 상기 제2 기술적 과제에 의한 선형 근사화를 이용한 애니메이션 궤적의 복원 장치에서 수행되는 애니메이션 궤적의 복원 방법을 제공하는 데 있다.

<32> 본 발명이 이루고자 하는 제5 기술적 과제는, 상기 제1 또는 상기 제3 기술적 과제에 의해 압축된 애니메이션 궤적의 압축된 포맷을 제공하는 데 있다.

【발명의 구성 및 작용】

<33> 상기 제1 과제를 이루기 위한 본 발명에 의한 선형 근사화를 이용한 애니메이션 궤적의 압축 장치는, 원래의 애니메이션 궤적으로부터 분절점들을 추출하는 보간 분석부와, 상기 보간 분석부로부터 출력되는 분절점들을 이용하여, 상기 보간 분석부로부터 입력한 키를 부호화하는 키 부호화부와, 상기 보간 분석부로부터 출력되는 분절점들을 이용하여, 상기 보간 분석부로부터 입력한 키값, 키값의 최소값 및 최대값들을 부호화하는 키값 부호화부 및 상기 키 및 키값 부호화부들에서 상기 부호화된 결과들을 엔트로피 부호화하여, 부호화된 비트 스트림으로서 출력하는 엔트로피 엔코더로 구성되는 것이 바람직하다.

<34> 상기 제2 과제를 이루기 위한 본 발명에 의한 선형 근사화를 이용한 애니메이션 궤적의 복원 장치는, 상기 부호화된 비트 스트림을 입력하여 엔트로피 디코딩하는 엔트로피 디코더와, 상기 엔트로피 디코딩된 결과를 입력하여 상기 키를 복호화하는 키 복호화부와, 상기 엔트로피 디코딩된 결과를 입력하여 상기 키값을 복호화하는 키값 복호화부 및 상기 키 및 상기 키값 복호화부에서 복호화된 결과를 입력하여 상기 원래의 애니메이션 궤적으로 복원하는 보간 복원부로 구성되는 것이 바람직하다.

<35> 상기 제3 과제를 이루기 위한 본 발명에 의한 선형 근사화를 이용한 애니메이션 궤적의 압축 방법은, 원래의 애니메이션 궤적으로부터 분절점들을 추출하는 단계와, 추출된 상기 분절점들을 이용하여, 키를 부호화하는 단계와, 추출된 상기 분절점들을 이용하여, 키값과 키값의 최소값 및 최대값들을 부호화하는 단계 및 상기 부호화된 키 및 키값들을 엔트로피 부호화하여, 부호화된 비트 스트림을 구하는 단계로 이루어지는 것이 바람직하다.

<36> 상기 제4 과제를 이루기 위한 본 발명에 의한 선형 근사화를 이용한 애니메이션 궤적의 복원 방법은, 상기 부호화된 비트 스트림을 엔트로피 디코딩하는 단계와, 상기 엔트로피 디코딩된 결과로부터 구한 상기 키를 복호화하는 단계와, 상기 엔트로피 디코딩된 결과로부터 구한 상기 키값을 복호화하는 단계 및 복호화된 상기 키 및 상기 키값을 상기 원래의 애니메이션 궤적으로 복원하는 단계로 이루어지는 것이 바람직하다.

<37> 상기 제5 과제를 이루기 위한 본 발명에 의한 선형 근사화를 이용한 애니메이션 궤적의 압축 및 복원 방법 및 그 방법들을 수행하는 장치들을 위한 부호화

된 비트 스트림은 상기 키의 개수, 상기 키의 해상도, 상기 키값의 해상도, 배열 형태의 상기 키값의 최대값 및 최소값, 배열 형태의 키 플래그, 배열 형태의 키 및 배열 형태의 키값으로 이루어지는 것이 바람직하다.

<38> 이하, 본 발명에 의한 선형 근사화를 이용하여 애니메이션 궤적을 압축 및 복원 장치들 각각의 구성 및 동작과, 그 장치들에서 수행되는 압축 및 복원 방법들 및 이를 위한 압축된 데이터 포맷을 첨부한 도면들을 참조하여 다음과 같이 설명한다.

<39> 본 발명에서는 VRML/MPEG-4에서 사용되고 있는 인터폴레이터 표현을 이용하며, 인터폴레이터가 사용되는 분야는 온 라인 컴퓨터 게임, 애니메이션 광고등이다.

<40> 도 5a는 본 발명에 의한 애니메이션 궤적의 압축 장치의 블럭도로서, 보간 분석(interpolator analysis)부(80), 키(K:Key) 부호화부(key coder)(82), 키 값(KV:Key Value) 부호화부(84) 및 엔트로피 엔코더(Entropy encoder)(86)로 구성되고, 도 5b는 본 발명에 의한 애니메이션 궤적의 복원 장치의 블럭도로서, 엔트로피 디코더(90), 키 복호화부(92), 키값 복호화부(94) 및 보간 복원(interpolator reconstruction)부(96)로 구성된다.

<41> 먼저, 도 5a에 도시된 압축 장치에서 보간 분석부(80)는 인터폴레이터(Interpolator) 분석(Analysis) 단계에서 부호화할 키(Key)와 키 값(Key Value)을 선정하게 된다. 이때 선택할 수 있는 방법은 각 종류별 인터폴레이터의 특성을 고려하여 설계할 수 있다. 분석의 예는 다음과 같이 후술된다.

- <42> 도 6은 도 5a에 도시된 압축 장치의 본 발명에 의한 바람직한 일실시예의 블럭도로서, 정규화(normalization)부(100) 및 분절점(break point) 최소화(minimum)부(102)로 구성되는 보간 분석부(80), 차분 펄스 부호 변조(DPCM:Differential Pulse Coding Modulation) 양자화부(104)로 구성되는 키 부호화부(82), DPCM 양자화부(106)로 구성되는 키값 부호화부(84) 및 엔트로피 엔코더(86)로 구성된다.
- <43> 도 7은 도 5b에 도시된 복원 장치의 본 발명에 의한 바람직한 일실시예의 블럭도로서, 엔트리피 디코더(90), DPCM 역 양자화부(126)로 구성되는 키 복호화부(92), DPCM 역 양자화부(128)로 구성되는 키값 복호화부(94) 및 키 & 키값 재건부(95)로 구성되는 보간 복원부(96)로 구성된다.
- <44> 먼저, 보간 분석부(80)의 입력으로 제공되는 인터플레이터(Interpolator) 표현은 각각 키(K:key)와 키값(KV:key value)로 구분하여 처리한다. 먼저, 도 6에 도시된 압축 장치에서 첫 단계인 보간(Interpolator) 분석부(80)는 입력단자 IN3을 통해 입력된 애니메이션 궤적을 최소의 분절점으로 표시할 수 있도록 분절점의 수를 조절한다. 원래 입력된 분절점의 수가 N개라면, 보간 분석부(80)를 거쳐 나오는 분절점의 수는 $M(M \leq N)$ 으로 조정된다. 즉, 보간 분석부(80)는 입력단자 IN3을 통해 입력한 원래의 애니메이션 궤적으로부터 분절점들을 추출하는 역할을 한다.
- <45> 인터플레이터의 경우, 키(key)와 키값(key value)은 각각 정규화(Nomalization)하여 사용할 수 있다. 이를 위해, 정규화부(100)는 입력단자 IN3을 통해 입력한 원래의 애니메이션 경로에 키 및 키값을 각각 정규화하여 분절점

최소화부(102)로 출력한다. 정규화는 서로 다른 범위로 분포되어 있는 값들의 처리를 일괄적으로 처리할 수 있다는 장점이 있다. VRML에서 지원하는 key는 0에서 1사이의 값을 갖는다.

<46> 보간 분석부(80)가 분절 점의 수를 조정하는 방식은 조정된 분절점이 만드는데 애니메이션 궤적과 원 애니메이션 궤적의 차이를 최소화하는 방향으로 결정될 수 있다. 이를 위해, 분절점 최소화부(102)는 정규화부(100)로부터 출력되는 정규화된 키 및 키 값으로부터 분절점들을 최소한으로 추출한다. 예를 들어, 위치 인터플레이터의 경우, 실제 궤적과 보간 분석부(80)를 통해 결정되는 양자화된 궤적사이에 가장 작은 면적 오차를 발생하도록 분절 점을 결정한다.

<47> 한편, 도 6에 도시된 분절점 최소화부(102)에서 수행되는 선형 근사화를 이용한 분절점 추출에 대해 살펴보면 다음과 같다.

<48> 도 8a-8h들은 선형 근사화를 이용한 분절점 추출의 본 발명에 의한 바람직한 일실시예를 설명하기 위한 도면들로서, 애니메이션 궤적상에 위치하는 각 원은 분절점을 나타낸다. 도 8a는 원 애니메이션의 궤적을 나타내고, 도 8b는 양단점(A, B)를 찾는 것을 보여주고, 도 8c는 원 궤적과 가장 가까운 분절점 선택을 보여주고, 도 8d는 1차 추출된 분절점 A, B, C를 보여주고, 도 8e는 원 궤적과 가까운 2차 분절점 추출을 보여주고, 도 8f는 2차 추출된 분절점 A, B, C 및 D들을 보여주고, 도 8g는 3차 추출된 분절점 A, B, C, D, E를 보여주고, 도 8h는 4차 추출된 분절점 A, B, C, D, E, F를 각각 보여준다.

<49> 실제 애니메이션 궤적이 도 8a와 같이 주어졌을 때, 양단점 A, B를 시작점으로 출발하여 각 분절점 중 원 궤적과의 차이가 가장 적은 근사화 궤적을 나타

내는 분절점을 찾는 과정을 반복한다. 이러한 과정은 도 8b ~ 8h를 통해 알 수 있다. 이 때, 분절점 추출은 근사화 궤적이 충분히 원궤적에 가깝다고 판단할 수 있을 때까지 반복할 수 있다. $2 \leq M \leq N$ 인가를 판단하는 근거로는 실제 궤적과 근사화 궤적의 면적 차이, 추출된 분절점의 수 등이 사용될 수 있다.

<50> 이하, 근사화 궤적과 실제 궤적의 차이를 구하는 방법으로 면적차를 이용하는 방법에 대해서 첨부한 도면을 참조하여 다음과 같이 설명한다.

<51> 도 9는 실제 애니메이션 궤적(200)과 근사화된 애니메이션 궤적(210)의 차이를 구하는 방법을 설명하기 위한 도면이다.

<52> 도 9를 참조하면, 궤적들은 다음 수학식 1과 같은 사다리꼴 또는 다음 수학식 2와 같은 비틀린 사다리꼴의 집합으로 표시할 수 있고, 이들의 면적의 합은 다음 수학식 3과 같은 두 궤적의 면적차(D_A)가 된다.

<53>
$$D_{\text{사다리꼴}} = \frac{(a+b)h}{2}$$

【수학식 1】

<54>
$$D_{\text{비틀린_사다리꼴}} = \frac{1}{2} \frac{a(3b-a)h}{a+b}, \quad b \geq a$$

【수학식 2】

<55>
$$D_A = \sum_i D_{\text{사다리꼴}} + \sum_j D_{\text{비틀린_사다리꼴}}$$

【수학식 3】

<56> 수학식 3에 표현된 면적차(D_A)를 가장 작게하는 방향으로 분절점을 추출하면 된다. 이상에서와 같이, 분절점 최소화부(102)는 분절점을 추출한다.

<57> 도 6에 도시된 보간 분석부(80)를 거치면 애니메이션 궤적의 필수 분절점 이

추출된다. 무 손실 처리가 필요할 경우, 추출된 분절점의 갯수(M)는 원래의 분절점의 수와 같을 수 있다. 위치 인터플레이터의 경우, 키 값이 X, Y, Z축의 삼차원 공간상의 이동을 표시하기 때문에 애니메이션 궤적은 X, Y, Z상의 3개의 곡선의 형태를 띤다. 보간 분석부(80)는 각 축에 독립적으로 분절점 추출을 수행할 수 있고, 이 때, 각 축별 분절점은 서로 같지 않을 수 있다. 다음과 같은 표 2는 8개의 분절점(P0, P1, P2, P3, P4, P5, P6 및 P7)을 갖는 실제 궤적을 보간 분석부(80)를 거쳐 새로운 분절점을 추출한 결과를 나타낸다.

<58> 【표 2】

분절점	P0	P1	P2	P3	P4	P5	P6	P7
kv _x	0	X	X	X	0	X	X	0
kv _y	0	X	0	X	0	X	X	0
kv _z	0	X	0	X	X	0	X	0
key	0	X	0	X	0	0	X	0
key_flag	7	-	6	-	3	4	-	7

<59> 표 2에서, key_flag는 키 플래그를 나타내고, kv_x, kv_y 및 kv_z는 좌표축 X, Y 및 Z에 대한 키값을 각각 나타낸다. 또한, '0'은 해당값을 보내지 않은 것을 의미하고, 'X' 및 '-' 각각은 해당값을 하나라도 보내지 않은 것을 의미하고, '7'은 모든 값들이 보내지는 것을 의미하고, '6'은 값 kv_x을 보내지 않은 것을 의미한다.

<60> 표 2에서 처럼 분절점 P1, P3, P6는 X, Y, Z축 궤적들 중 단 하나의 축에서도 선택되지 않아 불필요한 분절점으로 분류되었다. P0, P7은 양단점으로 모든 축에서 분절점으로 추출된 경우이고, P5처럼 X축에서는 사용되지 않지만, 다른 축에서는 사용되는 분절점들도 있다. 따라서 보간 분석부(80)에서는 원래 8개의

분절점 대신 5개의 분절점에 해당하는 키(Key)를 결정하고, 각 축별로 선정된 분절점에 해당하는 키 값(Key Value)을 보내게 된다. 표 2는 키(Key)가 5개, 키 값(Key Value)이 11개가 필요한 경우이다. 해당 Key와 Key Value의 대응관계는 키 플래그(key_flag)를 별도로 전송하여 표현할 수 있다. 이러한 표현 방식으로 키(Key)와 키 값(Key Value)의 개수는 실제 32개에서 16개와 부속 키 플래그(key_flag)로 줄어들게 된다. 이 과정을 통해 표현상의 간소화가 일어나게 된다.

<61> 도 14a-14f는 선형 근사화를 이용한 분절점 추출의 본 발명에 의한 바람직한 또 다른 일 실시예를 설명하기 위한 도면들로서, 회전 인터포레이터(Orientation Interpolator)에서 키프레임(key frame)(또는 분절점)을 줄이는 방법에 관한 것이다.

<62> 도 14a는 원래의 애니메이션 경로 상에서 $n+1$ 개의 시간에 따른 각 키프레임에서의 키값($= Q_0, Q_1, Q_2, \dots, Q_n$)를 보여주고 있으며, 검정색 점으로 표시되어 있다. 도 14b에서 보는 바와 같이 먼저 애니메이션 경로 상에 있는 각각의 키 프레임 중에서 양 끝에 해당하는 키 프레임 2개($= Q_0, Q_n$)를 선택한다. 선택된 점은 흰색으로 표시되어 있다.

<63> 도 14c는 선택된 양 끝 키 프레임을 제외한 나머지 키 프레임들 중에서 하나의 키 프레임을 선택하는 것을 보여준다. 이 때 하나의 키 프레임을 선택하는 방법은 모두 $n-1$ 가지가 있으며 도 14c에는 예시로 두 가지의 후보($= Q_1, Q_k$)를 선택한

모습을 보여주고 있다. 다음으로 $n-1$ 개의 후보에 대해서, 선택되지 않은 키 프레임의 키값들을 선택된 총 3개의 키 프레임들을 이용하여 구면 선형 보간 (spherical linear interpolation)에 의해서 보간한다. 원래의 애니메이션 경로와 보간된 $n-1$ 개의 애니메이션 경로를 비교하여 애니메이션 궤적의 오차가 가장 작은 애니메이션 경로를 선택하고 이로부터 $n-1$ 개의 후보 키 프레임으로부터 선택된 애니메이션 경로에 대응하는 키 프레임을 선택한다. 궤적간의 오차는 평균 오차 E_m 를 이용하여 구한다. 일례로 도 14d는 후보 2의 궤적이 선택되었음을 보인다.

<64> 도 14e에서와 같이, 선택된 3개의 키 프레임을 제외한 나머지 키 프레임들 중에서 하나의 키 프레임을 선택한 다음으로 위에서 설명된 도 14c와 관련된 과정을 수행한다. 일례로 도 14f는 후보 1의 궤적이 선택되었음을 보인다. 도 14e와 관련된 과정을 평균 오차가 허용 오차 보다 작아질 때까지 반복한다.

<65> 위에서 언급한 평균 오차 E_m 은 다음과 같이 구할 수 있다. 양자화 오차는 원래의 회전 변환과 복원된 회전 변환의 차분 회전 변환에서 차분 회전각으로 정의된다, 즉, 회전 인터포레이터 노드에 속한 하나의 키값을 (\vec{r}, θ) , 이를 복호화기에서 복원한 키값을 (\vec{r}', θ') 라 할 경우(단, 벡터 \vec{r} 은 회전축을, θ 는 회전량을 나타내며 그 범위는 $\theta \in [-\pi, \pi]$ 를 만족한다), (\vec{r}, θ) 와 (\vec{r}', θ') 에 의해 3차원 공간상의 임의의 위치 \vec{x} 로부터 \vec{y} 및 \vec{y}' 로의 회전 변환이 이루어질 때, 발생하는 양자화 오차는 \vec{y} 와 \vec{y}' 의 차로 계산된다. 이는 양자화 오차 벡터

$\vec{e}(\vec{x}) = \vec{y} - \vec{y}'$ 의 관계를 성립시킨다. $\vec{x}, \vec{y}, \vec{y}'$ 를 쿼터니온(quaternion)으로 표현하면 다음과 같이 정의된다.

<66>

$$\begin{cases} X = (0, \vec{x}) \\ Y = (0, \vec{y}) \\ Y' = (0, \vec{y}') \end{cases}$$

【수학식 4】

<67> 회전변환을 나타내는 (\vec{r}, θ) 와 (\vec{r}', θ') 를 쿼터니온으로 표현한 것을 각각 Q 와 Q' 라 할 경우, 다음의 관계식을 유도할 수 있다.

<68>

$$\begin{aligned} Y &= Q * X * Q^* \\ \text{【수학식 5】 } X &= Q^* * Y * Q \end{aligned}$$

<69> 여기서, $A*B$ 는 쿼터니온 곱셈을 나타내며, A^* 는 A 의 공액복소수(conjugate)를 의미한다. 이 관계식에 따라 다음의 식을 얻는다.

<70> 【수학식 6】 $Y' = Q' * X * Q'^* = Q' * Q^* * Y * Q * Q'^* = Q'' * Y * Q''^*$

<71> Q 는 \vec{y} 와 \vec{y}' 간 회전변환 관계를 나타내는 값으로 다음의 식과 같이 정의된다.

<72> 【수학식 7】 $Q'' = Q' * Q^*$

<73> 따라서, \vec{y} 와 \vec{y}' 간의 차분 회전각을 θ'' 라 하면 쿼터니온 변환식과 수학식 7에 의해 θ'' 는 다음과 같이 구해질 수 있다.

<74> 【수학식 8】 $\theta'' = 2\cos^{-1}q_0'' = 2\cos^{-1}(Q' \cdot Q), \theta'' \in [0, \pi], q_0'' = Q' \cdot Q$

<75> 여기서,

· 는 내적 연산을 의미한다. 수학식 8은 전체 애니메이션 키 프레임들 중 어느 특정 시간에 나타나는 순시 양자화 오차를 나타내며, 이를 전체 애니메이션 구간에 대한 양자화 오차를 구하는 식으로 유도하기 위해 특정 시간 t 에서의 순시 양자화 오차로 나타내면 다음과 같다.

<76> 【수학식 9】 $e(t) = 2\arccos(Q(t) \cdot Q'(t))$

<77> 수학식 9를 회전 인터포레이터가 애니메이션을 수행하는 전체 키 프레임 구간에 대해 확장하여 전체 구간 $[t_0, t_L]$ 에 대한 평균 에러 E_m 및 최대 에러 E_p 를 유도하면 다음과 같다.

<78>
$$E_m = \sqrt{\frac{1}{t_L - t_0} \int_0^L e^2(t) dt}$$

【수학식 10】 $E_p = \max|e(t)|$

<79> 여기서, E_m 을 구하기 위해 구간 $[t_{i-1}, t_i]$ 에서 부분합 E_m^i 을 먼저 구한다.

<80> 【수학식 11】 $E_m^i = \int_{t_{i-1}}^t e^2(t) dt = 4 \int_{t_{i-1}}^t \arccos^2(Q(t) \cdot Q'(t)) dt$

<81> 한편, 다음과 같은 식이 성립한다.

<82> 【수학식 12】 $4\arccos^2(Q(t) \cdot Q'(t)) = \Phi^2(\alpha), t = t_{i-1} + \alpha(t_i - t_{i-1})$

<83> 따라서,

<84> 【수학식 13】 $E_m^i = (t_i - t_{i-1}) \int_0^1 \Phi^2(\alpha) d\alpha$

<85> 적분구간 $[0, 1]$ 에서 함수 $\Phi^2(\alpha)$ 의 정적분을 구하기 어려우므로, 다음과 같이 근사화 한다.

<86>

$$\Phi(\alpha) \cong \Phi(0) + \alpha(\Phi(1) - \Phi(0))$$

$$\Phi^2(\alpha) \cong \Phi^2(0) + \alpha^2(\Phi(1) - \Phi(0))^2 + 2\alpha\Phi(0)(\Phi(1) - \Phi(0))$$

【수학식 14】

<87>

여기서,

<88>

$$\cos \frac{\Phi(0)}{2} = Q(t_{i-1}) \cdot Q'(t_{i-1}), \quad \cos \frac{\Phi(1)}{2} = Q(t_i) \cdot Q'(t_i)$$

【수학식 15】

<89>

근사화 된 함수를 사용하여 부분합 E_m^i 을 구하면 다음과 같다.

<90>

$$E_m^i \cong \frac{1}{3} (t_i - t_{i-1})(\Phi^2(0) + \Phi^2(1) + \Phi(0)\Phi(1))$$

【수학식 16】

<91>

위 식들을 다시 정리하면, 다음과 같다.

<92>

$$E_m^i \cong \frac{4}{3} (t_i - t_{i-1})(\arccos^2(Q(t_{i-1}) \cdot Q'(t_{i-1})) + \arccos^2(Q(t_i) \cdot Q'(t_i)) \\ + \arccos(Q(t_{i-1}) \cdot Q'(t_{i-1})) \arccos(Q(t_i) \cdot Q'(t_i)))$$

【수학식 17】

<93>

부분합 E_m^i 를 전체 구간 $[t_0, t_L]$ 에 대해 합하여 평균 에러 E_m 을 구한다.

<94>

$$E_m \cong \sqrt{\frac{1}{t_L - t_0} \sum_{i=0}^L E_m^i}$$

【수학식 18】

<95>

최대 에러 E_p 는 각 구간 $[t_{i-1}, t_i]$ 에서의 최대 에러 E_p^i 를 다음과 같이 구한 후, 그 중 최대값을 구하면 된다.

<96>

$$E_p^i = \max |e(t)| = \max 2 |\arccos(Q(t) \cdot Q'(t))|$$

【수학식 19】

<97>

앞서 사용한 근사함수를 이용하면,

<98>

$$E_p^i \cong \max(\Phi(0), \Phi(1)) = \max \{2|\arccos(Q(t_{i-1}) \cdot Q'(t_{i-1}))|, 2|\arccos(Q(t_i) \cdot Q'(t_i))|\}$$

【수학식 20】

<99>

전체 구간 $[t_0, t_L]$ 에서 구한 최대값은,

<100> 【수학식 21】 $E_p \cong \max E_p^i, \quad \text{for } i = 1, 2, \dots, L$

<101> 도 6에 도시된 보간 분석부(80)는 키의 개수, 정규화된 키 값의 최소치와 최대치, 키의 해상도, 키값의 해상도 및 키 플래그등의 정보(114)를 엔트로피 엔코더(86)로 보낸다. 이 때, 보간 분석부(80)는 키(110)를 키 부호화부(82)로 보내고, 정규화된 키 값의 최소치와 최대치 및 키값에 대한 정보(112)를 키값 부호화부(84)로 보낸다.

<102> 한편, 키 및 키값 부호화부들(82 및 84)에서 수행되는 키 및 키값 부호화에 대해서 다음과 같이 설명한다.

<103> 키 및 키 값은 각각 키 부호화부(82)와 키 값 부호화부(84)를 통해 DPCM 방식으로 양자화 압축된다. 양자화된 정보는 기타 정보들과 함께 엔트로피 엔코더(86)를 거쳐 최종 압축되어 부호화된 비트 스트림으로서 출력단자 OUT3을 통해 출력된다. 예컨대, 키 부호화부(82)의 DPCM 양자화부(104)는 보간 분석부(80)로부터 입력한 키를 DPCM 양자화 방식으로 부호화하고, 즉, 바로 전의 값과 현재 값의 차이를 부호화하고, 부호화된 결과를 엔트로피 엔코더(86)로 출력한다. 이 때, 키값 부호화부(84)의 DPCM 양자화부(106)는 보간 분석부(80)로부터 입력한 키값 및 키값의 최소값 및 최대값들을 DPCM 양자화 방식으로 부호화하고, 즉, 바로 전의 값과 현재 값의 차이를 부호화하고, 부호화된 결과를 엔트로피 엔코더(86)로 출력한다.

<104> 도 10은 양자화 과정을 나타내는 도면으로서, 일반적으로 사용되는 DPCM 부호화 방식을 나타낸다.

<105> 도 10을 참조하면, 키(K:Key)의 경우 다음 수학적식 4와 같이 최종 오차(e)를 양자화 결과로 압축한다. 즉, 도 10은 다음 수학적식 4를 설명하는 도면이다.

<106>

$$\begin{aligned} K_i &= 0, \text{ for } i < 0 \\ d1 &= K_{i-1} - K_{i-2}, \\ \widehat{K}_i &= K_{i-1} + d1, \\ d2 &= K_i - \widehat{K}_i, \\ \text{【수학적식 22】 } e &= d2 \end{aligned}$$

<107> 도 11a-11e들은 본 발명에 의한 부호화된 비트 스트림의 포맷을 나타내는 도면이다.

<108> 도 11a는 본 발명에 의한 부호화된 비트 스트림의 포맷으로서, 키의 갯수 (n_key)와, 키의 해상도(k_res), 키값의 해상도(kv_res), 키값의 최소값 및 최대값([min/max])(여기서, []는 배열을 의미함)(230), 키 플래그 ([key_flag])(232), 키([key])(234) 및 키 값([kv])(236)으로 구성된다. 이 때, 도 11b는 도 11a에 도시된 키값의 최소값 및 최대값(230)의 본 발명에 의한 포맷을 나타내고, 도 11c는 키 플래그(232)의 포맷을 나타내고, 도 11d는 키(234)의 포맷을 나타내고, 도 11e는 키값(236)의 포맷을 각각 나타낸다.

<109> 예컨데, 도 11a에 도시된 배열 형태의 키값의 최대값 및 최소값(230)는 도 11b에 도시된 바와 같이, X, Y 및 Z 좌표축에 대한 최소값들(min_x, min_y 및 min_z) 및 X, Y 및 Z 좌표축에 대한 최대값들(max_x, max_y 및 max_z)로 구성된다.

<110> 도 11a에 도시된 배열 형태의 키 플래그(232)는 n-1(여기서, n은 키의 개수를 나타낸다.)개의 키 플래그들(key_flag₀, key_flag₁, key_flag₂, ... 및 key_flag_{n-1})로 구성된다. 도 11a에 도시된 배열 형태의 키(234)는 n-1개의 키

들($key_0, key_1, key_2, \dots$ 및 key_{n-1})로 구성된다. 도 11a에 도시된 배열 형태의 키값(236)은 $p(\leq n)$ 개의 제1 키값들(kv_X_0, \dots, kv_X_p), $q(\leq n)$ 개의 제2 키 값들(kv_Y_0, \dots, kv_Y_q) 및 $z(\leq n)$ 개의 제3 키값들(kv_Z_0, \dots 및 kv_Z_r)로 구성된다.

<111> 한편, 본 발명에 의한 애니메이션 복원 장치 및 방법은 압축 과정의 역순으로 다음과 같이 설명된다.

<112> 도 7에 도시된 엔트로피 디코더(90)는 입력단자 IN4를 통해 입력한 부호화된 비트 스트림을 입력하여 엔트로피 디코딩하고, 엔트로피 디코딩된 결과를 키 복호화부(92) 및 키값 복호화부(94)로 각각 출력한다. 키 및 키값 복호화부들(92 및 94)은 각각 엔트로피 디코딩된 결과인 키 및 키 값이 양자화된 결과를 입력하여, 복호화된 부가정보와 함께 양자화되기 전의 결과로 복원시킨다. 즉, 키 복호화부(92)는 엔트로피 디코딩된 결과를 입력하여 키를 복호화하고, 키값 복호화부(94)는 엔트로피 디코딩된 결과를 입력하여 키값을 복호화한다. 이를 위해, 키 복호화부(92)는 DPCM 역 양자화부(126)로 구현될 수 있고, 키값 복호화부(94)는 DPCM 역 양자화부(128)로 구현될 수 있다. 각 DPCM 역 양자화부(126 또는 128)는 엔트로피 디코더(90)로부터 입력한 디코딩된 결과를 DPCM 역 양자화 방식으로 복호화한다.

<113> 상기 키 및 상기 키값 복호화부에서 복호화된 결과를 입력하여 상기 원래의 애니메이션 궤적으로 복원하는 보간 복원부

<114> 이 때, 보간 복원부(96)는 엔트로피 디코더(90)로부터 키의 개수, 정규화된 키 값의 최소치와 최대치, 키의 해상도, 키값의 해상도 및 키 플래그등의 정보

(124)를 입력하고, 키 복호화부(92)로부터 키(120)를 입력하고, 키값 복호화부(94)로부터 정규화된 키 값의 최소치와 최대치 및 키값에 대한 정보(122)를 입력하고, 비어있는 키 값을 선형 삽입으로 복원하고, 복원된 결과를 복원된 애니메이션 경로로서 (키, 키값) 형태로 출력단자 OUT4를 통해 출력한다. 이를 위해, 보간 복원부(96)는 키 & 키값 제건부(95)로 구현될 수 있다. 예를 들면 표 2에서 분절점 P2의 X축 키값은 분절점 P0, P4를 선형 삽입과정을 통해 복원할 수 있다.

<115> 도 12는 문법표현으로 바꾸어 표현한 본 발명에 의한 복호화 처리 과정을 설명하기 위한 도면으로서, 비트 스트림 신택스(syntax)이다.

<116> 본 발명에 의한 선형 근사화를 이용한 애니메이션 궤적의 압축 방법과 종래의 MPEG-4 BIFS를 이용한 인터플레이터 압축 방법을 첨부한 도면들을 참조하여 다음과 같이 설명한다.

<117> 도 13a-13d들은 본 발명에 의한 압축 방법과 종래의 압축 방법을 비교하기 위한 실험 애니메이션 시퀀스(sequence)에 대한 그래프들로서, 종축은 왜곡도를 나타내고, 횡축은 비트수를 각각 나타내며, 본 발명에 의한 애니메이션 궤적의 압축 방법은 굵은 실선으로 표시되었고 종래의 압축 방법은 점선으로 표시되었다. 도 13a ~ 13d들은 각기 다른 애니메이션 궤적을 적용할 때의 시뮬레이션 결과들을 나타낸다.

<118> 도 13b-13a-13d-13c의 순서로 갈수록 좋은 결과를 보여줌을 알 수 있다. 도

13a ~ 13d들을 참조하면, 동일한 비트율에서 본 발명에 의한 압축 방법은 종래의 BIFS 방법보다 월등한 화질을 나타내고, 같은 화질에서 월등한 압축율을 보여주고 있다.

【발명의 효과】

<119> 이상에서 설명한 바와 같이, 본 발명에 의한 선형 근사화를 이용한 애니메이션 궤적의 압축 및 복원 장치들과 이 장치들에서 수행되는 압축 및 복원 방법들 및 이를 위한 데이터 포맷은, 애니메이션 궤적의 인터플레이터 표현의 핵심은 분절점을 통한 궤적의 단순화임에도 불구하고 종래에는 궤적의 표현이 수작업 또는 자동 작업을 통해 획득되는 과정에서 키의 간격이 일정하게 형성되기 때문에 많은 경우 분절점들은 과샘플링(Oversampling)되는 것과 대비하여 볼 때, 분절점 분석을 통해 필요한 분절점을 줄여서 부호화된 비트 스트림을 단순화시킬 수 있는 효과, DPCM을 통한 압축에 의해 단순화된 분절점들간에도 선형 상관성을 높게 유지시킬 수 있고 이러한 높은 상관성을 효과적으로 압축하여 주는 효과들을 갖는다.

【특허청구범위】**【청구항 1】**

원래의 애니메이션 궤적으로부터 분절점들을 추출하는 보간 분석부;

상기 보간 분석부로부터 출력되는 분절점들을 이용하여, 상기 보간 분석부로부터 입력한 키를 부호화하는 키 부호화부;

상기 보간 분석부로부터 출력되는 분절점들을 이용하여, 상기 보간 분석부로부터 입력한 키값, 키값의 최소값 및 최대값들을 부호화하는 키값 부호화부;
및

상기 키 및 키값 부호화부들에서 상기 부호화된 결과들을 엔트로피 부호화하여, 부호화된 비트 스트림으로서 출력하는 엔트로피 엔코더를 구비하는 것을 특징으로 하는 선형 근사화를 이용한 애니메이션 궤적의 압축 장치.

【청구항 2】

제1 항에 있어서, 상기 보간 분석부는

상기 원래의 애니메이션 궤적으로부터 상기 키 및 상기 키 값을 정규화하여 출력하는 정규화부; 및

상기 정규화된 키 및 키 값으로부터 상기 분절점들을 최소한으로 추출하는 분절점 최소화부를 구비하는 것을 특징으로 하는 선형 근사화를 이용한 애니메이션 궤적의 압축 장치.

【청구항 3】

제2 항에 있어서, 상기 분절점 최소화부는 조정된 상기 분절점이 만드는 애니메이션의 궤적과 상기 원래의 애니메이션 궤적의 차이를 최소화하는 방향으로 상기 분절점의 수를 결정하는 것을 특징으로 하는 선형 근사화를 이용한 애니메이션 궤적의 압축 장치.

【청구항 4】

제1 항에 있어서, 상기 키 부호화부는 상기 보간 분석부로부터 입력한 키를 차분 펄스 부호 변조(DPCM) 양자화 방식으로 부호화하고, 상기 DPCM 양자화 방식은 바로 전의 값과 현재 값의 차이를 부호화하는 것을 특징으로 하는 선형 근사화를 이용한 애니메이션 궤적의 압축 장치.

【청구항 5】

제1 항에 있어서 상기 키값 부호화부는 상기 보간 분석부로부터 입력한 키값 및 키값의 최소값 및 최대값들을 상기 DPCM 양자화 방식으로 부호화하고, 상기 DPCM 양자화 방식은 바로 전의 값과 현재 값의 차이를 부호화하는 것을 특징으로 하는 선형 근사화를 이용한 애니메이션 궤적의 압축 장치.

【청구항 6】

제3 항에 있어서, 상기 분절점 최소화부는

상기 원래의 애니메이션 궤적의 양 단점을 시작점으로 하여, 각 분절점들중 서로간의 차이가 가장 적은 근사화 궤적을 나타내는 분절점을 찾고, 이러한 과정을 근사화된 애니메이션 궤적이 상기 원래의 애니메이션 궤적과 가까워질 때까지

지 반복하는 것을 특징으로 하는 선형 근사화를 이용한 애니메이션 궤적의 압축 장치.

【청구항 7】

제4 항 또는 제5 항에 있어서, 상기 DPCM 양자화 방식을 사용할 때, 바로 이전 값과 그 이전값의 차이를 바로 이전값에 더한 값과 현재 값의 차이를 부호화하는 것을 특징으로 하는 선형 근사화를 이용한 애니메이션 궤적의 압축 장치.

【청구항 8】

제7 항에 있어서, 상기 분절점 최소화부는

상기 원래의 애니메이션 궤적과 상기 양자화된 궤적 사이에 가장 작은 면적 오차를 발생하도록 분절점을 결정하는 것을 특징으로 하는 선형 근사화를 이용한 애니메이션 궤적의 압축 장치.

【청구항 9】

제1 항에 있어서,

상기 부호화된 비트 스트림을 입력하여 엔트로피 디코딩하는 엔트로피 디코더;

상기 엔트로피 디코딩된 결과를 입력하여 상기 키를 복호화하는 키 복호화부;

상기 엔트로피 디코딩된 결과를 입력하여 상기 키값을 복호화하는 키값 복호화부; 및

상기 키 및 상기 키값 복호화부에서 복호화된 결과를 입력하여 상기 원래의 애니메이션 궤적으로 복원하는 보간 복원부를 구비하는 것을 특징으로 하는 선형 근사화를 이용한 애니메이션 궤적의 복원 장치.

【청구항 10】

제9 항에 있어서, 상기 키 복호화부는 상기 엔트로피 디코더로부터 입력한 디코딩된 결과를 차분 펄스 부호 변조(DPCM) 역 양자화 방식으로 복호화하는 것을 특징으로 하는 선형 근사화를 이용한 애니메이션 궤적의 복원 장치.

【청구항 11】

제9 항에 있어서 상기 키값 복호화부는 상기 엔트로피 디코더로부터 입력한 디코딩된 결과를 DPCM 역 양자화 방식으로 복호화하는 것을 특징으로 하는 선형 근사화를 이용한 애니메이션 궤적의 압축 장치.

【청구항 12】

원래의 애니메이션 궤적으로부터 분절점들을 추출하는 단계;

추출된 상기 분절점들을 이용하여, 키를 부호화하는 단계;

추출된 상기 분절점들을 이용하여, 키값과 키값의 최소값 및 최대값들을 부호화하는 단계; 및

상기 부호화된 키 및 키값들을 엔트로피 부호화하여, 부호화된 비트 스트림을 구하는 단계를 구비하는 것을 특징으로 하는 선형 근사화를 이용한 애니메이션 궤적의 압축 방법.

【청구항 13】

제12 항에 있어서,

상기 부호화된 비트 스트림을 엔트로피 디코딩하는 단계;

상기 엔트로피 디코딩된 결과로부터 구한 상기 키를 복호화하는 단계;

상기 엔트로피 디코딩된 결과로부터 구한 상기 키값을 복호화하는 단계;

및

복호화된 상기 키 및 상기 키값을 상기 원래의 애니메이션 궤적으로 복원하는 단계를 구비하는 것을 특징으로 하는 선형 근사화를 이용한 애니메이션 궤적의 복원 방법.

【청구항 14】

제1 항, 제9 항, 제12 항 또는 제13 항중 어느 항에 있어서, 상기 부호화된 비트 스트림은

상기 키의 개수, 상기 키의 해상도, 상기 키값의 해상도, 배열 형태의 상기 키값의 최대값 및 최소값, 배열 형태의 키 플래그, 배열 형태의 키 및 배열 형태의 키값을 구비하는 것을 특징으로 하는 선형 근사화를 이용한 애니메이션 궤적의 압축 및 복원 방법 및 그 방법들을 수행하는 장치들을 위한 데이터 포맷.

【청구항 15】

제14 항에 있어서, 상기 배열 형태의 상기 키값의 최대값 및 최소값은

X, Y 및 Z 좌표축에 대한 최소값들; 및

X, Y 및 Z 좌표축에 대한 최대값들을 구비하는 것을 특징으로 하는 선형 근사화를 이용한 애니메이션 궤적의 압축 및 복원 방법 및 그 방법들을 수행하는 장치들을 위한 데이터 포맷.

【청구항 16】

제14 항에 있어서, 상기 배열 형태의 상기 키 플래그는 $n-1$ (여기서, n 은 상기 키의 개수를 나타낸다.)개의 키 플래그들을 구비하는 것을 특징으로 하는 선형 근사화를 이용한 애니메이션 궤적의 압축 및 복원 방법 및 그 방법들을 수행하는 장치들을 위한 데이터 포맷.

【청구항 17】

제14 항에 있어서, 상기 배열 형태의 키는 $n-1$ (여기서, n 은 상기 키의 개수를 나타낸다.)개의 키들을 구비하는 것을 특징으로 하는 선형 근사화를 이용한 애니메이션 궤적의 압축 및 복원 방법 및 그 방법들을 수행하는 장치들을 위한 데이터 포맷.

【청구항 18】

제14 항에 있어서, 상기 배열 형태의 키값은 $p(\leq n)$ 개의 제1 키값들, $q(\leq n)$ 개의 제2 키값들 및 $z(\leq n)$ 개의 제3 키값들을 구비하는 것을 특징으로 하는 선형 근사화를 이용한 애니메이션 궤적의 압축 및 복원 방법 및 그 방법들을 수행하는 장치들을 위한 데이터 포맷.

【청구항 19】

(a) 원래의 애니메이션 경로 상에서 시간 구분에 따른 각 키프레임에서의 키값을 설정하는 단계;

(b) 애니메이션 경로 상에 있는 각각의 키 프레임 중에서 양 끝에 해당하는 키 프레임 2개를 선택하는 단계;

(c) 상기 선택된 양 끝 키 프레임을 제외한 나머지 키 프레임들 중에서 하나의 키 프레임을 선택하는 단계;

(d) 선택되지 않은 키 프레임의 키값들을 선택된 키 프레임들을 이용하여 보간하는 단계;

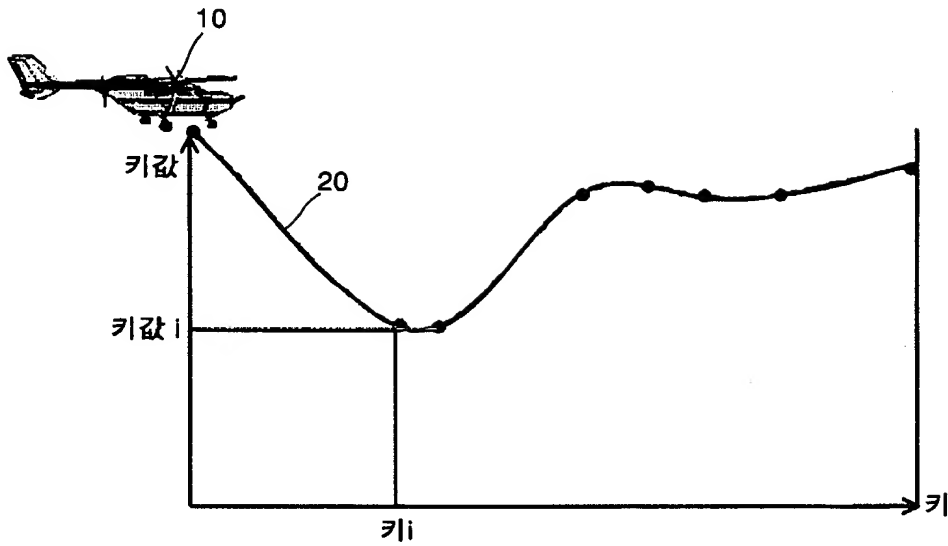
(e) 원래의 애니메이션 경로와 보간된 애니메이션 경로를 비교하여 애니메이션 궤적의 오차가 가장 작은 애니메이션 경로를 선택하고 후보 키 프레임으로부터 선택된 애니메이션 경로에 대응하는 키 프레임을 선택하는 단계;

(f) 선택된 키 프레임을 제외한 나머지 키 프레임들 중에서 하나의 키 프레임을 선택한 다음 상기 (c) 단계부터 다시 수행하는 단계; 및

(g) 상기 (f) 단계를 상기 궤적 간의 오차가 허용 오차 보다 작아질 때까지 반복하는 단계를 포함하는 것을 특징으로 하는 선형 근사화를 이용한 분절점 추출방법.

【도면】

【도 1】



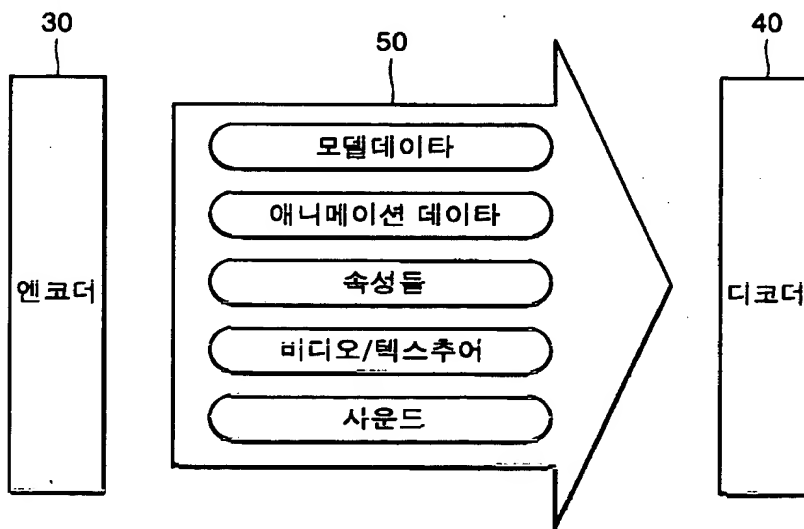
【도 2】

```

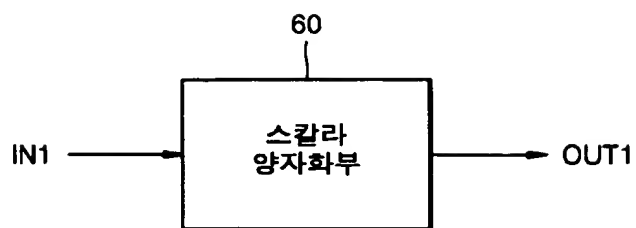
ScalarInterpolator {
    eventIn SFFloat set_fraction
    exposedField MFFloat key [ ]
    exposedField MFFloat keyValue [ ]
    eventOut SFFloat value_changed
}

```

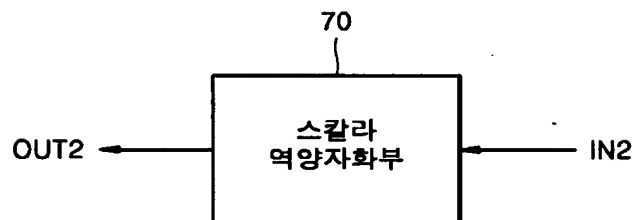
【도 3】



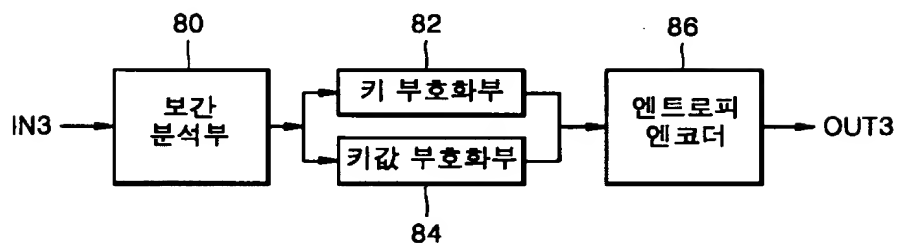
【도 4a】



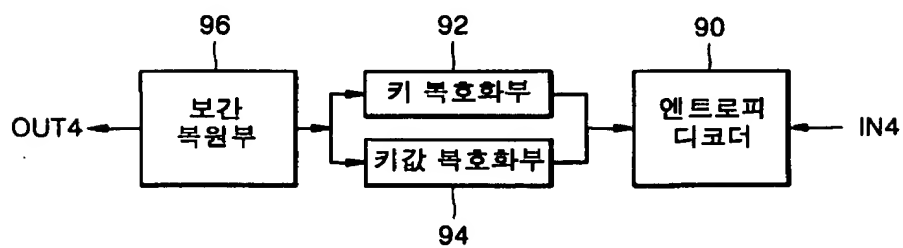
【도 4b】



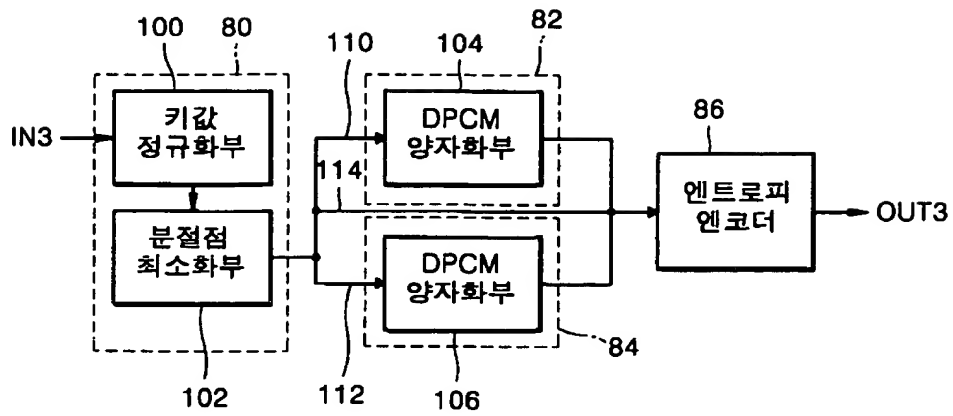
【도 5a】



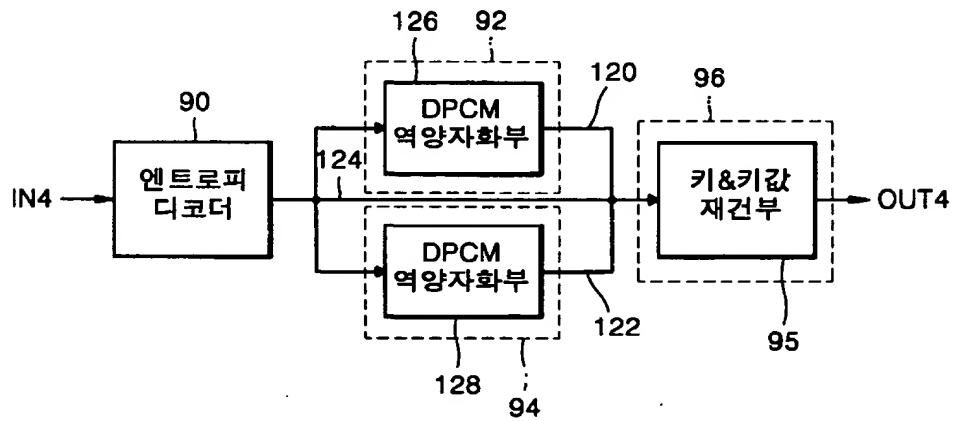
【도 5b】



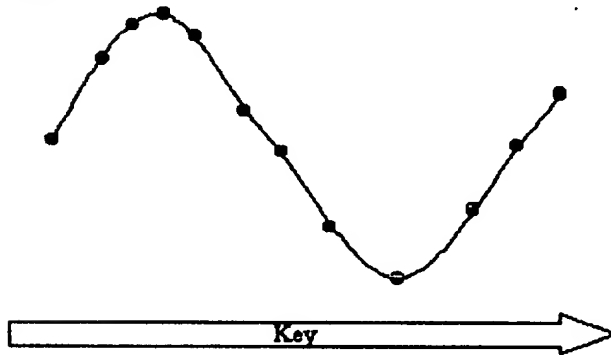
【도 6】



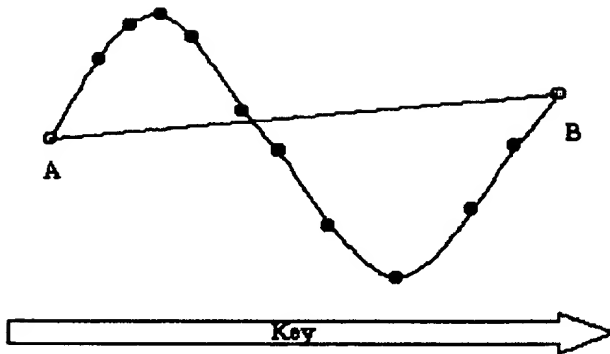
【도 7】



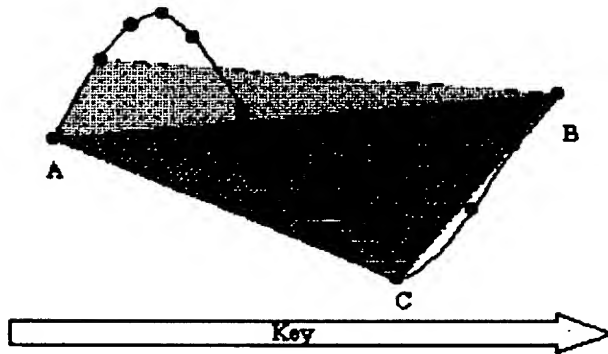
【도 8a】



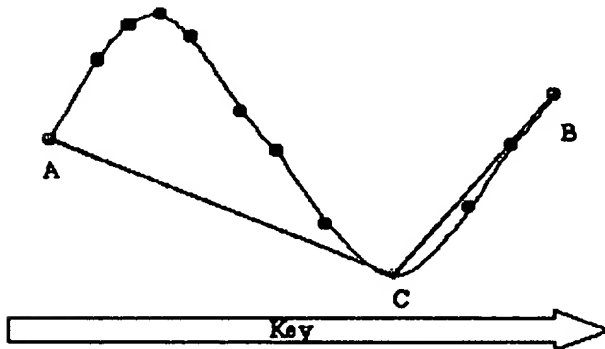
【도 8b】



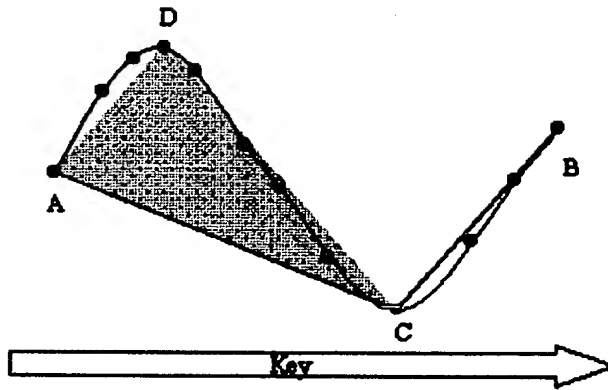
【도 8c】



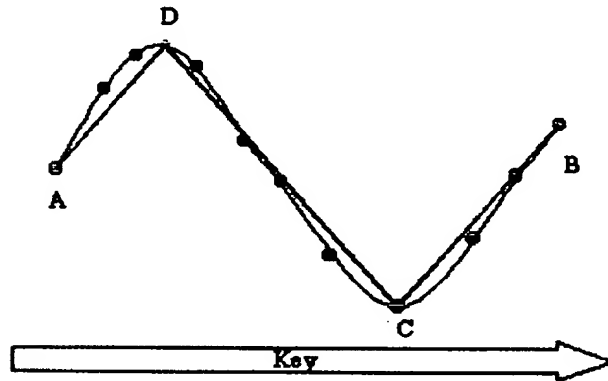
【도 8d】



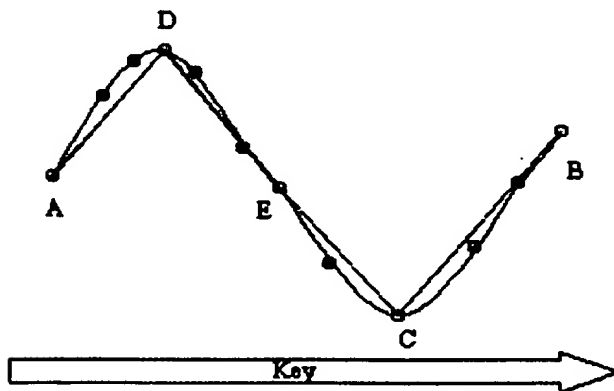
【도 8e】



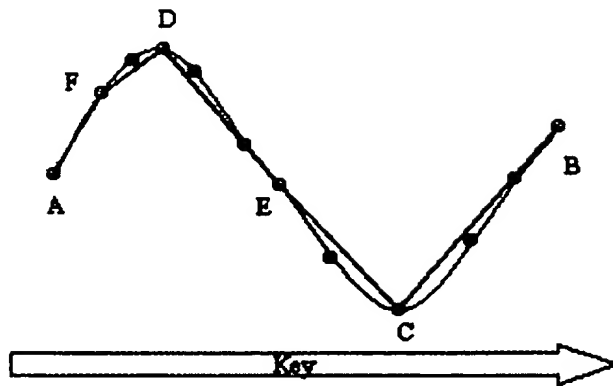
【도 8f】



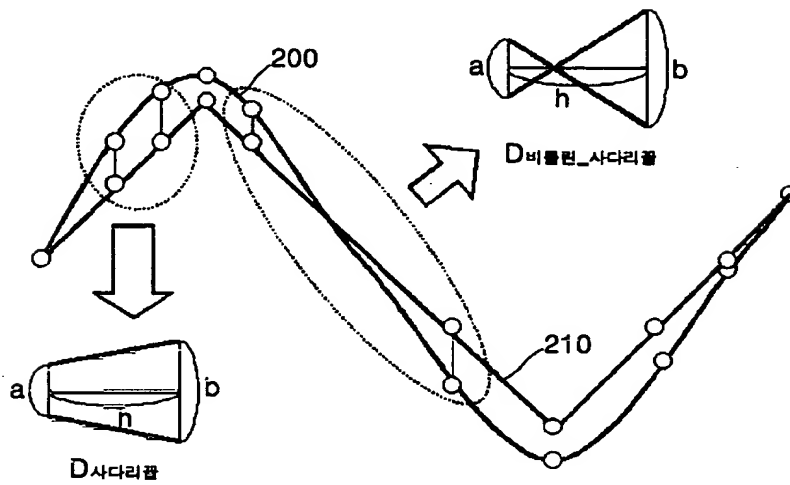
【도 8g】



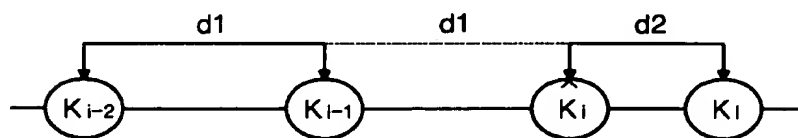
【도 8h】



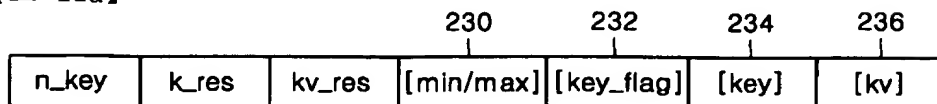
【도 9】



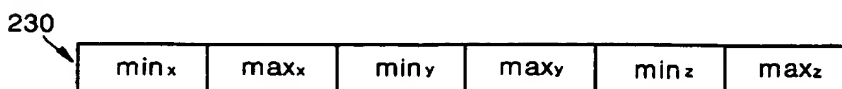
【도 10】



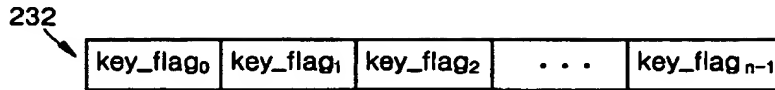
【도 11a】



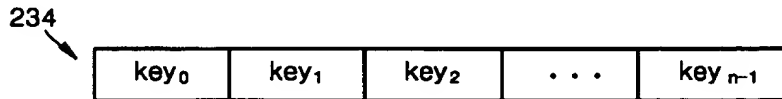
【도 11b】



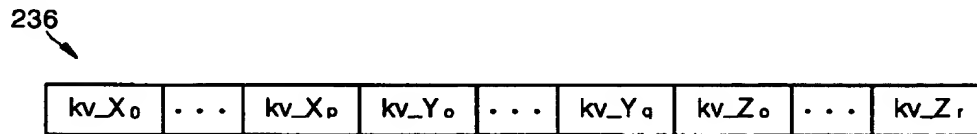
【도 11c】



【도 11d】



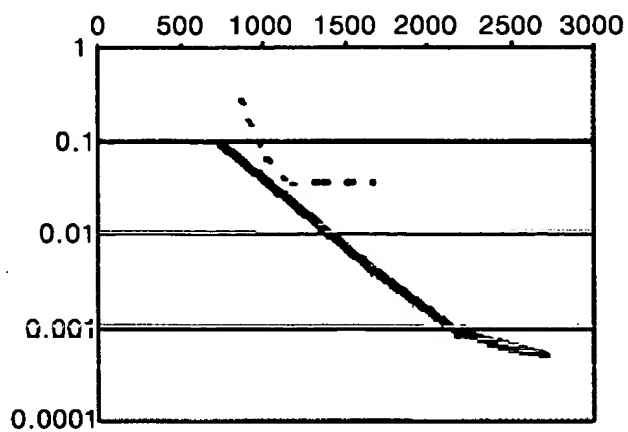
【도 11e】



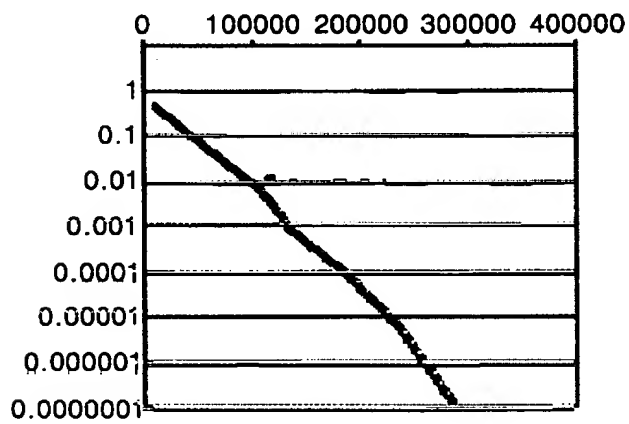
【도 12】

CompressedPositionInterpolator()	No. of bits	Mnemonic
{		
decodeAAC_start();		
n_key	16	uimstbf
k_res	16	uimstbf
kv_res	16	uimstbf
min_x	16	uimstbf
max_x	16	uimstbf
min_y	16	uimstbf
max_y	16	uimstbf
min_z	16	uimstbf
max_z	16	uimstbf
for (i=0; i<n_key, i++)		
{		
decodeAAC(key_flag[i], Key_flag_Context);		viclbf
}		
for (i=0; i<n_key, i++)		
{		
decodeAAC(key[i], Key_Context);		viclbf
}		
decipher_key_flag(key_flag);		
for (i=0; i<n_kv_x, i++)		
{		
decodeAAC(kv[i], KV_Context);		viclbf
Inv_Quantize(kv[i]);		
}		
for (i=0; i<n_kv_y, i++)		
{		
decodeAAC(kv[i], KV_Context);		viclbf
Inv_Quantize(kv[i]);		
}		
for (i=0; i<n_kv_z, i++)		
{		
decodeAAC(kv[i], KV_Context);		viclbf
Inv_Quantize(kv[i]);		
}		
decodeAAC_finish();		
}		

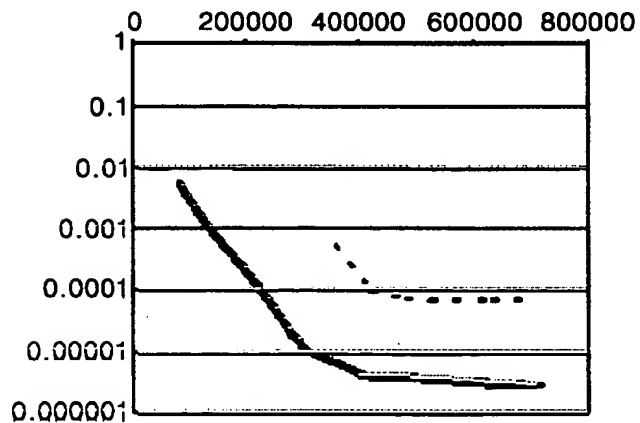
【도 13a】



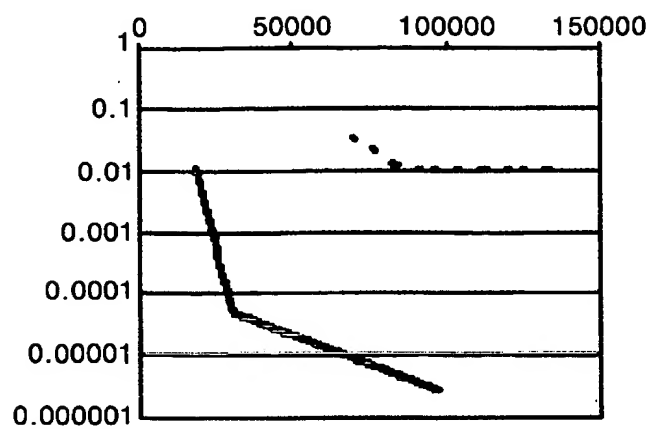
【도 13b】



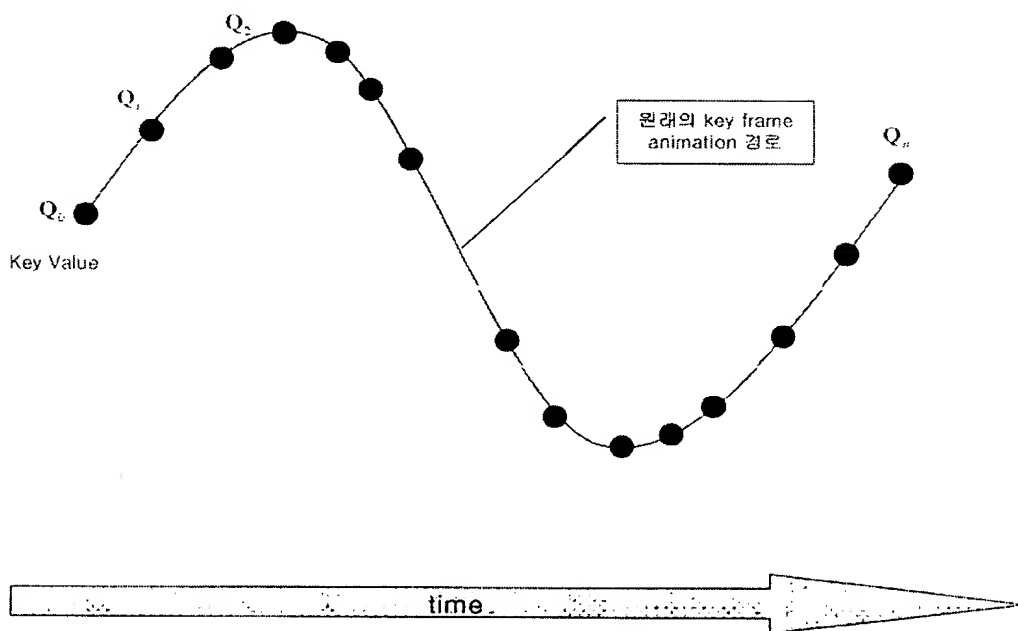
【도 13c】



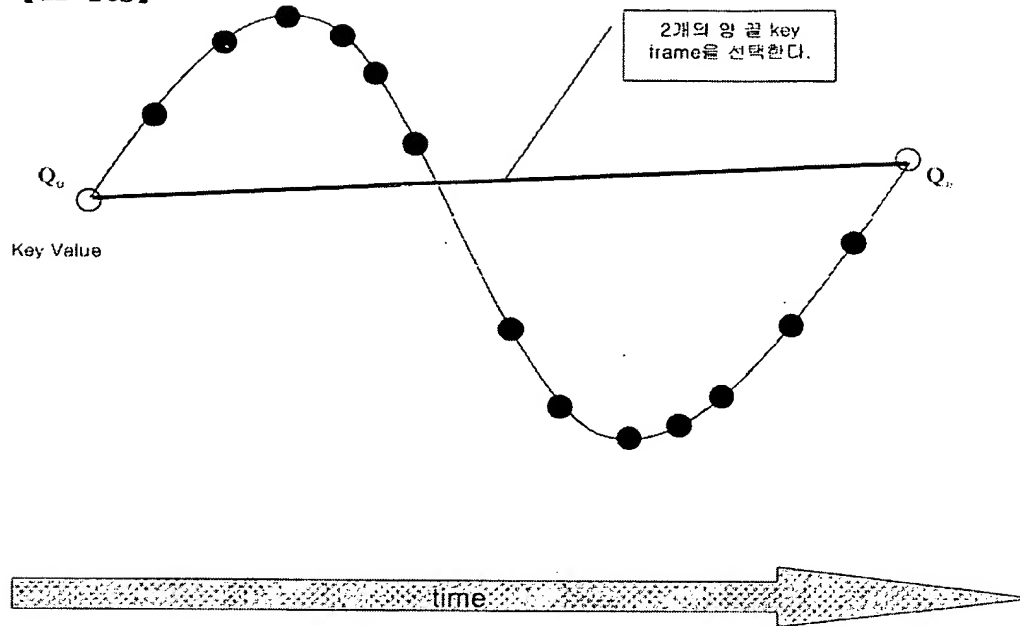
【도 13d】



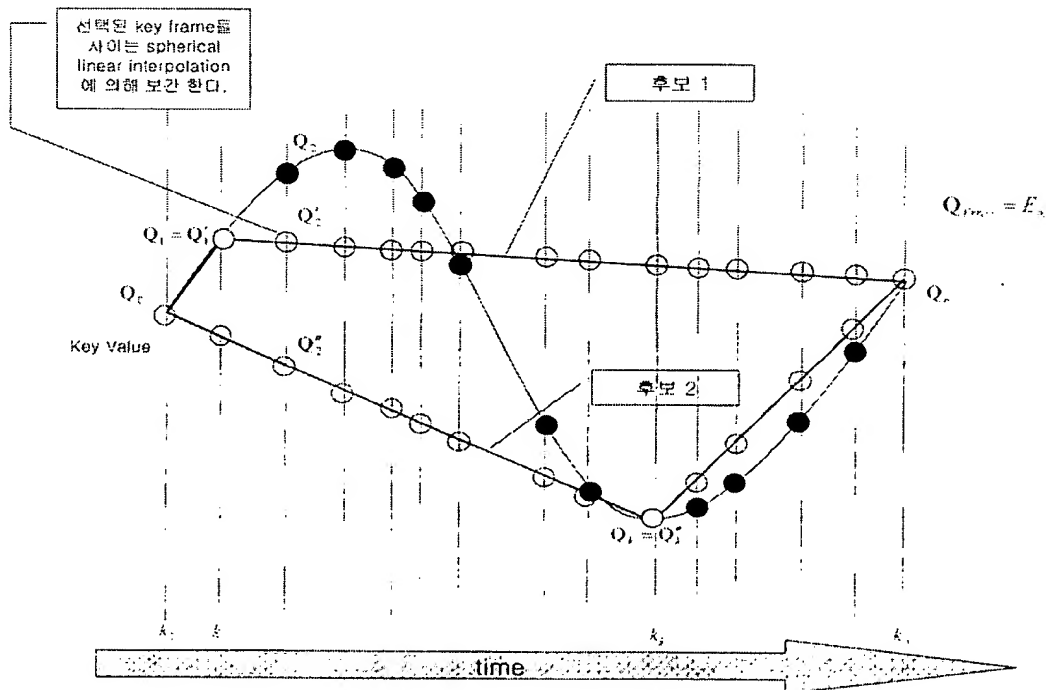
【도 14a】



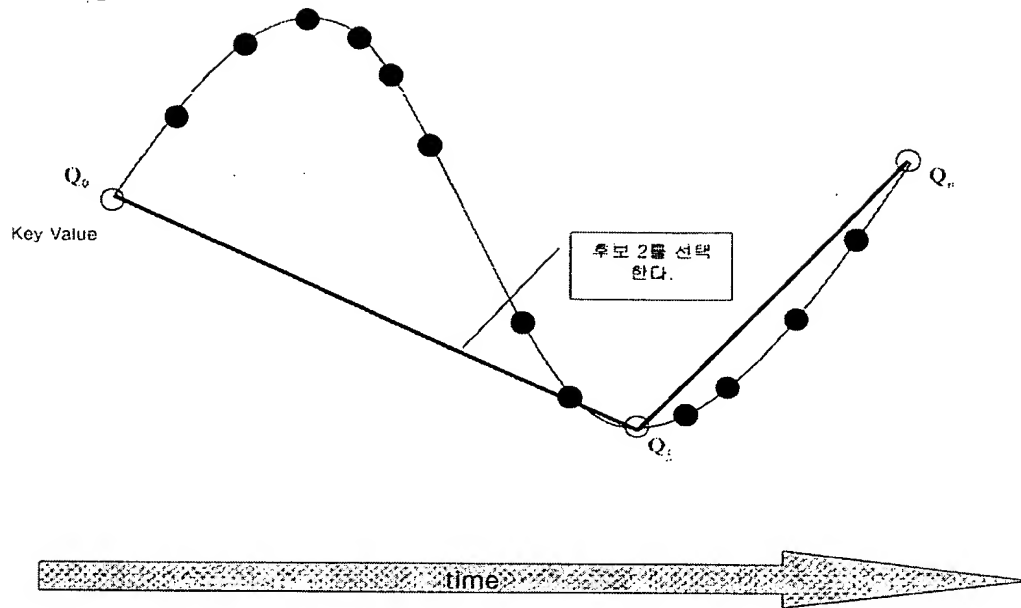
【도 14b】



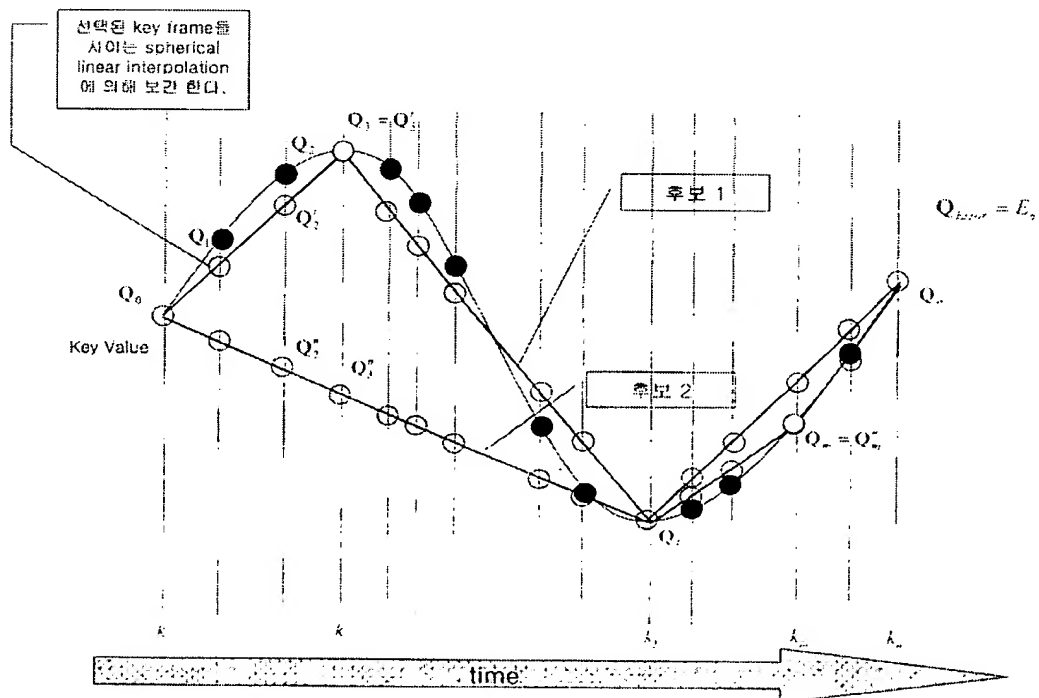
【도 14c】



【도 14d】



【도 14e】



【도 14f】

